

REMARKS

This Amendment and Response responds to the Office Action dated January 17, 2006.

Section 1: Response To “Response To Arguments”:

In response to the detailed response to the arguments that were presented in the Submission, reference is made to Subsections 1.1-1.2 below.

Subsection 1.1: Response To Response To Arguments On Page 3 Of The Action: In Paragraph 4 of the Action, it was indicated that full consideration had been given to the remarks against Chung U.S. Patent 6105148 (Chung), but that such remarks were found not persuasive.

Subsection 1.1.A:

The Response To Arguments on page 3 of the Action focused on one phrase used in the Submission, namely “levels of importance”. The page 3 response indicated that:

“the feature of ‘replication of data, based on methods and apparatus which establish levels of importance of replicating data’ is not clearly established in the plain language of the claims.”

The Response To Arguments required the claims to indicate which state management type is more important, in that the response stated:

“...it is not clear which state management type (i.e., disk-replicated or memory-replicated) is more important than which” (emphasis added).

In response, in the previous Submission remarks the claimed replication of data was said to be “based on methods and apparatus which establish levels of importance of replicating data for restorative purposes”. It is respectfully submitted that this remark was a summary way of referring to the effect of the disclosed “policy” that is identified by the claimed state management type for replication of the claimed state objects to the claimed dedicated (different types of) state servers. As such, the noted “levels of importance” need not appear, in so many words, in the claims. Rather, the claims set forth the operations and system by which the “policy” that may be identified by the claimed state management type is used replicate the

claimed state objects to the claimed dedicated types of state servers, and conform to the specification. For example, that “policy” is initially described at page 7 as follows:

...the present invention fills these needs by providing state management units that allow application developers to **specify the mechanism and policies of replicated state management** to be used with the entity beans of the application....

That “policy” is further described at page 7 as follows:

The application developer is allowed to classify individual entity beans with a particular state management type. Then, during execution, a plurality of state objects are provided, where each state object stores the state of a corresponding entity bean object within the memory address space of a Java server process...

Further, each state object is associated with the state management type of the corresponding entity bean object. The state management type identifies the mechanism and policy for replication of state objects to the different types of state servers...

The different types of state management are described in the specification at page 7 as including:.

The state management type can be a memory replicated state management type, a disk replicated state management type, a non-replicated state manage type, or other state management type that may be needed for a [sic] entity bean.

It is observed that this recitation of state management types does not indicate which type is more important than another type. Instead, in the present invention it would be proper and possible for a developer to specify an exemplary policy as follows:

“The replication of every entity bean object having a disk replicated state management type will be to a disk replicated state server, which is a state server that corresponds to the disk replicated state management type”. The replication of every entity bean object having a memory replicated state management type will be to a memory replicated state server, which is a state server that corresponds to the memory replicated state management type”.

Such a policy would be consistent with Applicants’ specification. For example, starting at page 16, line 21, it is stated in reference to FIG. 3:

The RSM manages the recoverable state of entity bean 304 **based on the type of the state. Depending on the type of state, the RSM replicates the state** in either a disk replicated state server 212 or a memory replicated state server 214.

FIG. 3 clearly shows the “Disk Replicated State Server” 212 as a server that is separate from the “Memory Replicated State Server” 214. The quoted page 16 description of the RSM is

further explained at page 26, starting at line 13, at which the replication to these state servers 212 and 214 is described as:

More specifically, for each state partition 954 that participated in a particular transaction with a pending checkpoint, the RSM 206 delegates a Checkpoint manager 950 to issue the checkpoint, in operation 901. In response, the checkpoint manager 950 extracts the checkpoint state **from the memory replicated SMUs 310** for each SMU of memory replicated type, in operation 902. The Checkpoint manager 950 then sends the checkpoint **to the memory replicated state server 214**, in operation 903. In a similar manner, the checkpoint manager 950 extracts the checkpoint state **from the disk replicated SMUs 308** for each SMU of disk replicated type, in operation 904, and sends the checkpoint **to the disk replicated state server 212**, in operation 905.

It is again observed that this recitation of replication using the state management types does not indicate that any state management type must be, or is to be, more important than another type. Rather, it is clear from this disclosure, and from the claims based thereon, that for example there is a correspondence of the state associated with a particular entity bean to the state object that corresponds to that entity bean (Claim 9, “partitioning” clause), and to the particular state management unit that is a collection of the state objects corresponding to one particular state management type (claim 9, “classifying” clause). In the page 26, lines 13+ words, for a state object having a memory replicated state management type, the replication is from the memory replicated SMUs to the memory replicated state server. In the page 26, lines 13+ words, for a state object having a disk replicated state management type, the replication is from the disk replicated SMUs to the disk replicated state server. Moreover, consistent with that taught at the quoted page 26, claim 9 defines:

replicating each particular state management unit in one of a plurality of state servers according to the particular state management type that corresponds to the particular state objects classified in the particular state management unit.

In view of these remarks, it is respectfully submitted that the claims do not require any indication of the relative importance of the state management types. Accordingly, consideration of the claims as being proper without a definition of one state management type being more important than another is respectfully requested.

Subsection 1.1.B:

The Response To Arguments further asserted that the recoverable state does not necessarily have a “level of importance” that is different from that of the state management of the second type (the non-recoverable type). The reasons for this assertion were that (1) there

are only two possible state management types (disk-replicated or memory-replicated), and (2) the recoverable state is claimed as being either memory replicated or disk replicated. Independent claims 1 and 18 that define the recoverable state have been clarified to define the recoverable state as being one of memory and disk (claim 1, “classifying” clause; claim 18, “application” clause).

The Response To Arguments also asserted that:

Since the recoverable state management type is not claimed as being of an exclusive type (i.e., exclusively disk replicated, *or* exclusively memory-replicated) to clearly distinguish it from the non-recoverable state management type, the two different management types are not deemed to have different “levels of importance”.

It is respectfully submitted this further Response To Arguments is answered by the above remarks. In detail, as to “exclusive type”, reference is made to the quotes of page 16, lines 21+ and page 26, lines 13+. These quotes support the claims. Amended claim 1 defines:

providing state management for each entity bean object using a state object associated with the state management type corresponding to the respective entity bean object, the providing state management being based separately on each different state management type and on those state objects corresponding to the different state management type...

In the amended claim 9 “replicating” clause, there is replicating of a particular state management unit in one state server “according to the particular state management type that corresponds to the particular state objects classified in the particular state management unit”.

In claim 18, there is defined:

a replicated state manager configured to replicate a particular state management unit to the state server that is dedicated to the particular state management type of the particular state object that is classified into the particular state management unit to be replicated.

By the absence from these claims of the non-recoverable state management type, and by the noted claim text, the “exclusivity” claimed is that one state management unit stores state objects corresponding to only one state management type---and that one state management unit is replicated only to the state server that is dedicated to that one state management type.

Consideration of the claims, as may be amended, as being proper is respectfully requested.

Further, as to the assertion that the recoverable state does not necessarily have a “level of importance” that is different from that of the state management of the second type (the non-recoverable type), reference is made to the specification. The specification page 7 quote as to state management types does not indicate which type is more important than another type, thus the claims need not include such a recitation. Consideration of the claims, as may be amended, as being proper is respectfully requested.

Subsection 1.1.C:

The Response To Arguments went further, by asserting:

Rather, as claimed, there is no clear distinction between which state objects (hence entity bean objects) are to be memory replicated and which state objects are to be disk replicated since any recoverable state object can be replicated to memory or disk.

In response, as should be clear from the above remarks relating to the claims as may be amended, the claims do clearly distinguish between which state objects (hence entity bean objects) are to be memory replicated and which state objects are to be disk replicated. Further, the specification does not teach, thus the claims need not define, that **any** recoverable state object can be replicated to memory **or** disk.

In support, it is noted that the claim 1 clauses define:

wherein each state object is associated with the state management type of the **corresponding** entity bean object; and

providing state management for each entity bean object using a state object associated with the state management type corresponding to the respective entity bean object, the providing **state management** being based **separately** on each **different** state management type **and on those state objects** corresponding to the **different** state management type, the providing state management comprising replicating each one of the plurality of state objects in a state server, a **different one of the state servers being dedicated to a particular one of the state management types, a different one of the state servers being provided for each different recoverable state management type.**

Also, the classifying clause of amended claim 9, for example, defines:

wherein each particular state management unit is a collection of the state objects corresponding to **one particular state management type**

Further, claim 9 also sets forth:

replicating each **particular** state management unit in one of a plurality of state servers **according to the particular state management type** that corresponds to the **particular state objects classified in the particular state management unit**.

In addition, claim 18 defines:

a plurality of state management units that classify the state objects, a particular state object being classified into a particular state management unit based on the particular state management type...

a state server dedicated to each state management type...

a replicated state manager configured to replicate a particular state management unit to the state server that is dedicated to the particular state management type of the particular state object that is classified into the particular state management unit to be replicated.

It is respectfully submitted that as now clarified (claims 1 and 9) and as now claimed (claim 18), **any** one state server cannot be used to replicate **any** recoverable state object to memory **or** disk. Rather, there is a clear distinction in the (specification and) claims between how some state objects are to be memory replicated (for example), and how some state objects are to be disk replicated. This clear distinction is because the defined replicating is in a state server according to the different (or particular) state management type that corresponds to the particular state objects classified in the particular state management unit. Moreover, the replication is of the particular state management unit that collects only one particular state management type. Thus, the claims define that in an exemplary situation, if there are state objects that are memory replicated (and are thus collected in **one** state management unit), those state objects will be memory replicated (by being replicated in the memory state server).

In terms of claim 1, for example, this follows from providing state management by replicating each one of the plurality of state objects in a state server, a different one of the state servers being dedicated to a particular one of the state management types, and a different one of the state servers being provided for each different recoverable state management type.

In terms of claim 9, for example, this follows from the memory state server replication being according to the particular state management type (here, memory replicated) that corresponds to the particular state objects classified in the state management unit (i.e., classified in the memory replicated unit).

In terms of claim 18, for example, this follows from the state server dedicated to each state management type, the state management type identifying a policy for replication of a state object to a state server dedicated to a particular state management type; and the replicated state manager configured to replicate a particular state management unit to the state server that is dedicated to the particular state management type of the particular state object that is classified into the particular state management unit to be replicated.

The same applies to the disk replicated state objects.

In view of the comments above, because the state objects of any replicated state management unit are all of one particular state management type as derived from the original entity beans (of that one particular state management type), the replication of the state objects of one state management unit to an exemplary disk state server are a replication of the original entity beans (of that one disk replicated state management type), on which the disk replicated state objects were based.

In view of these remarks, it is respectfully submitted that the claims are clear as presented. Accordingly, consideration of the claims as being proper is respectfully requested.

Subsection 1.2: Response To Response To Arguments On Page 4 Of The Action: On page 4 of the Action, further remarks were presented in re Chung in response to Applicants' remarks in the Submission. Those further remarks relate to the rejection based on Nally in view of Chung, which rejection is no longer asserted.

Section 2: Response To New Rejections:

Subsection 2.1: New Grounds of Rejection: In view of Applicants' remarks in the Submission, new grounds of rejection have been asserted, with the previously-cited secondary reference to Apte et al. U.S. Patent 6269373 (Apte) now being a sole or primary reference. Thus, in the Action:

Paragraph 6 cited Apte et al. U.S. Patent 6269373 (Apte) as the basis for anticipation of claims 1, 4, and 5;

In Paragraph 8 claims 6-7, 9, 13-20, and 25 were rejected under 35 USC Section 103(a) based on Apte in view of Chung U.S. Patent 6105148 (Chung);

In Paragraph 9 claim 8 was rejected under 35 USC Section 103(a) based on Apte in view of Nally et al. U.S. Patent 6298478 (Nally);

In Paragraph 10 claim 21 was rejected under 35 USC Section 103(a) based on Apte in view of Chung in view of Savage et al. U.S. Patent 6604110 (Savage).

Thus, Apte, previously cited as a secondary reference for only the features of serializing transactions and asserted partitioning entity bean objects into state partitions via fields 1210-1214 (Actions mailed 8/12/04 and 5/16/05), has thus now been cited as the primary reference against all of the pending claims. In the following Sections 2.2.A – 2.2.D, cites to Apte from the above Paragraphs 6, and 8-10 are reviewed in re the assertions that the cites teach the claimed invention, and the cites are distinguished from the claims.

Subsection 2.2: Response To Paragraphs 6, and 8-10 Asserting Apte Against Independent Claims 1, 9, and 18:

Subsection 2.2.A:

Apte Does Not Teach The Claims 1 and 18 Recoverable State Being One Of Memory Type And Disk Type

In claims 1 and 18, this aspect of the recoverable state is defined as:

the state management type being one of a recoverable state or a non-recoverable state, the recoverable state being one of a memory replicated state management type and a disk replicated state management type;

The claim 1 rejection (Action page 5) of the recoverable state management types of the claim 1 clause cited “(...at least the 1222 *Back-end storage* FIG. 12 & associated text”).

The claim 18 rejection (Action page 12) of the recoverable state management types of the claim 18 clause cited “(...see at least 1108, 1112 FIG. 11 & associated text)”.

Considering FIG. 12, aside from the description of the drawings (C3, L33-35), FIG. 12 is described only at C17, L30-C18, L12. Thus, C17 and C18 must be the “associated text”. Considering C17 and C18, it is respectfully submitted that here Apte does not teach, as claimed, the recoverable state being one of a memory replicated state management type and a disk replicated state management type.

In detail, C17 does not refer to 1222 *Back-end storage*. Continuing to C18, C18 does not describe FIG. 12 of Apte teaching that the recoverable state is one of a memory replicated

state management type and a disk replicated state management type, as claimed. At C18, there is an erroneous reference (underlined below for emphasis) to “backend storage 1214” in the text that starts at C17, L66:

When an EJB that refers to another EJB needs to persist its state information, the Tie object performs the stringify operations for storing the referred EJB in the back-end data store. Tie object 1206 also retrieves the persisted object reference from backend storage 1214 to reconstruct object reference 1208 by destringifying the previously stored string.

The reference to “...storage 1214” apparently should be to “....backend storage 1222...”. Importantly, that reference to “backend storage 1214” (C18, L3), when read as “backend storage 1222”, does not indicate that the “backend storage” 1222 provides a recoverable state being one of a memory replicated state management type or a disk replicated state management type. To the contrary, there is no indication that only one state management type is replicated in storage 1222, and no indication that another storage 1222 is provided for only one different state management type.

In view of the use of “at least” in the rejection, which could indicate that the Examiner believes that there is an unidentified further asserted teaching of the claimed recoverable state management types of this clause of claim 1, Apte has been reviewed beyond the C17-C18 descriptions of FIG. 12. This review noted C2, L1-2, that indicates that:

“Most back-end stores allow the persistence of primitive data types....”

Clearly, this is not a teaching that the back-end stores provide a recoverable state being one of a memory replicated state management type and a disk replicated state management type.

Considering the cite to FIG. 11, the cited “1108” is the question of whether the argument represents an EJB (an EJB wrapped by a wrapper)? Also, the cited “1112” is a question “more arguments?” Considering FIG. 11, aside from the description of the drawings (C3, L31+), FIG. 11 is described only at C10, and C14-17. The description of the cited 1108 is at C14, L50+, which relates to (L51) whether the argument represents an EJB. If so, step 1110 of unwrapping the adapter occurs. Respectfully, this is not a description of the claimed recoverable state being one of a memory replicated state management type and a disk replicated state management type. Further, the cite to 1112 in FIG. 11 is also not a description of the claimed recoverable state being one of a memory replicated state management type and

a disk replicated state management type. The reason is that 1112 is the question as to whether there are more arguments to be processed for the method (L56-60).

Continuing with the review of FIG. 11, at C15, L 14+, reference is made to storing the state information of the server bean, which is not a state object based on the EJB (client object). The description then relates generally to “persistence” at C15, L18-27. However, no reference is made there to the claimed recoverable state comprising the claimed two state management types. Following this general discussion, an EJB server is described, the containers are described, and various other Java features are described, leading to a C16, L18+ discussion of stateful sessions, and container-managed persistence (C16, L57+ onto C17, L1-20). However, in this discussion of topics such as “flattening” and Tie objects, there is no teaching of the claimed two state management types. The discussion then proceeds to FIGs. 12-14.

Lacking an identification in the rejection of a further place in Apte of asserted teaching that the back-end stores provide a recoverable state being one of a memory replicated state management type and a disk replicated state management type, and in view of these remarks, it is respectfully submitted that Apte does not teach the claims 1, 4 and 5 recoverable state being one of a memory replicated state management type and a disk replicated state management type. Because every claimed aspect of the defined operations must be shown by the reference cited for anticipation, and because Apte is shown to be lacking this noted feature, withdrawal of the rejection of claims 1, 4, and 5 based on Apte is respectfully requested.

Lacking an identification in the rejection of a further place (beyond FIG. 11) in Apte of asserted teaching that 1108 and 1112 provide a recoverable state being one of a memory replicated state management type and a disk replicated state management type, and in view of these remarks, it is respectfully submitted that Apte does not teach the claims 9, 13-17, 18-21, and 25 recoverable state being one of a memory replicated state management type and a disk replicated state management type. Because Apte is shown to be lacking this noted feature, and because the combined references must show every claimed aspect of the defined operations to make a *prima facie* case of obviousness, withdrawal of the rejection of claims 9, 13-17, 18-21, and 25 under Section 103 (a) is respectfully requested.

Subsection 2.2.B:

Apte Does Not Teach The Claimed Providing Of A State Object Storing State in A Java Process Server As Defined In Claims 1 and 18

Claims 1 and 18 define the state object as follows:

each state object storing a state of a corresponding entity bean object within a memory address space of a Java server process,

The rejection (Action page 6) of this claim 1 clause cited “(see at least 1202 *server* FIG. 12 & associated text) ” , and the rejection (Action page 12) of this claim 18 clause cited “(see at least 1202 FIG. 12 & associated text) ”.

Considering FIG. 12, aside from the description of the drawings (C3, L33-35), FIG. 12 is described only at C17, L30-C18, L12. Thus, C17 and C18 must be the “associated text”. Considering C17 and C18, it is respectfully submitted that here Apte does not teach, as claimed, that state objects are provided, each state object storing a state of a corresponding entity bean object within a memory address space **of a Java server process**.

In detail, at C17, L30+, it is clearly indicated that (bold added):

More particularly, the present invention uses a stringified CORBA reference as a way to persist an EJB. In the description of FIGS. 12-14, **the EJBs run on a CORBA-compliant server**. At runtime, an EJB reference is a CORBA proxy wrapped by an adapter class, as described above with respect to FIGS. 7-11. The method of the present invention implements a Tie class that provides communication between an EJB and a **CORBA-compliant server**. The TIE object is responsible for transferring data between an EJB and the server that hosts it.

With reference now to FIG. 12, a block diagram depicts system components for enabling JavaBeans as container-managed fields **of a CORBA server** according to the method of the present invention. **Server 1202** has container-managed EJB 1208 within container 1204, which manages the state information for EJB 1208. EJB has container-managed fields 1210, 1212, and 1214.

It is clear from this passage of Apte that the cited server 1202 shown in FIG. 12 is not the claimed “Java server process”, but instead is the CORBA-compliant server that gave rise to the problem with which Apte deals (C17, L55-65). Further, without a Java server process in Apte, any state objects that might store a state (represented by flags associated with fields 1210, or 1212, or 1214—see C17, L61-64) of an entity bean object, do not store a state of a

corresponding entity bean object within a memory address space of a Java server process, as claimed. Therefore, it is respectfully submitted that Apte does not teach, as claimed, that state objects are provided, each state object storing a state of a corresponding entity bean object within a memory address space **of a Java server process**.

Because every claimed aspect of the defined operations must be shown by the reference cited for anticipation, and because Apte is shown to be lacking this noted feature, withdrawal of the rejections of claims 1, 4 and 5 based on anticipation by Apte is respectfully requested. As to claims 9, 13-17, 18-21, and 25, because the above remarks show Apte does not teach this claimed feature, and because the combined references must show every claimed aspect of the defined operations to make a *prima facie* case of obviousness, withdrawal of the rejection of claims 9, 13-17, 18-21, and 25 under Section 103 (a) is respectfully requested.

Subsection 2.2.C:

Apte Does Not Teach The Claim 1, 9 and 18 State Server For A Particular State Object Being Dedicated To A Particular State Management Type

The rejection of the claim 1 clause included a reference to the claim 1 state server text, now clarified to read:

a different one of the state servers being **dedicated to a particular** one of the state management types, a **different one** of the state servers being provided for each different recoverable state management type.

In re claim 1, the cite (Action page 7, top) to Apte was to “...at least *1202 server, 1204 Container, 1206 Tie, 1208 EJB* FIG. 12 & associated text”.

The rejection of the claim 9 clause included a reference to the claim 9 replicating text that includes state servers, now clarified to read:

replicating each particular state management unit in **one** of a plurality of state servers **according to the particular** state management type that corresponds to the particular state objects classified in the particular state management unit.

In re claim 9, the cite (Action page 10) to Apte for plural servers was to “(see at least *server 104* FIG. 1 and associated text, *additional servers* col. 3: 58-col. 4:5)”; and for replicating according to the particular state management type (corresponding to the state objects in a state management unit) was to “(see at least *EJBs, protocol, particular server, mechanisms, persistent, container* col. 7: 25-55)”

The rejection of the claim 18 clause included a reference to the claim 18 state server and replicated state manager text, which reads:

a state server **dedicated** to each state management type, the state management type identifying a policy for replication of a state object to a state server dedicated to a particular state management type and a policy for migration of a state object from one server process to another server process; and

a replicated state manager configured to replicate a **particular** state management unit to the state server that is **dedicated** to the **particular** state management type of the particular state object that is classified into the particular state management unit to be replicated.

In re claim 18, the same cite (Action pages 12 & 13) was made to Apte for both the state management units and dedicated state servers, namely: “(see at least *EJBs, protocol, particular server, mechanisms, persistent, container* col. 7: 25-55)”.

Consideration is next given to each of these cites.

Subsection 2.2.C.1: Discussion of The Cite To:

“...at least 1202 server, 1204 Container, 1206 Tie, 1208 EJB FIG. 12 & associated text”.

Initially, it is respectfully submitted that as clarified by the above amendments to claims 1 and 9, and as to claim 18, in describing the 1202 server, 1204 Container, 1206 Tie, and 1208 EJB of FIG. 12, Apte does not teach the claimed state servers, replicating and state management units. **The reason is that, for example, Apte does not teach the claimed dedication of a state server to a different (or particular) state management type, and does not teach the related state management units for each particular type, for example.**

In support, reference is made to the remarks in Section 2.2.B in re the description of FIG. 12 at C17, L30-C18, L12 of 1202 server. In connection with 1204 Container, 1206 Tie, and EJB 1208, it was shown in Section 2.2.B that the server 1202 is a CORBA server on which the EJBs within containers 1204 run. Server 1202 is thus not a state server. This means that the FIG. 12 showing of what is included in 1202 server is not a showing of a dedication of a state server to a different (or particular) state management type.

Rather, the description is of the one EJB 1208 shown in FIG. 12 in the container 1204. L45-46 state “EJB has container-managed fields 1210, 1212, and 1214.” FIG. 12 shows these fields in the EJB 1208, not in a state management unit (e.g., Applicants’ FIG. 3) grouped or

classified with other same fields. Thus, FIG. 12 and this C17 description do not teach any claimed state management unit, for example, into which only one state management type of state objects are grouped; nor many such state management units for many state management types.

Moreover, the reference at C17, L58-60 to the Tie object 1206 does not amount to grouping of many of the fields 1214 of many EJBs via the claimed state objects all having the same state management type, nor to a state management unit into which those state objects having the same state management type are located, nor the claimed replicating into state servers, each dedicated only to one state management type. Instead, the Tie object 1206 described at C17, L58+ is used as part of deployment of EJB 1208 to assist in persistence of EJB 1208.

Further, C17, L66 onto C18, describes the function of the Tie object 1206 as performing the stringify operation, and not an operation of a replicated state manager configured to replicate a particular state management unit to the state server that is dedicated to the particular state management type of the particular state object that is classified into the particular state management unit to be replicated (e.g., claim 18). Also, at C17, L17+ the Tie object function of flattening a server reference is not such operation of the claimed replicated state manager, nor of the claim 9 replicating each particular state management unit in one of a plurality of state servers according to the particular state management type that corresponds to the particular state objects classified in the particular state management unit. Finally, these Tie object descriptions are not descriptive of the claim 1 particular one of the state servers being dedicated to a particular one of the state management types, a different particular one of the state servers being provided for each different recoverable state management type.

As to “associated text” in this cite in this Subsection 2.2.C.1., i.e., (“...1208 EJB FIG. 12 & associated text”), the associated text in re FIG. 12 continues onto C18. As noted in Subsection 2.2A., C18 does not describe FIG. 12 of Apte teaching that the recoverable state is one of a memory replicated state management type and a disk replicated state management type, as claimed. At C18, the erroneous reference to “backend storage 1214”, when read as “backend storage 1222”, does not indicate that the “backend storage” 1222 comprises separate state servers, one dedicated to each of many state management types, for the following reasons.

First, one, not plural, back-end storage 1222 is shown in FIG. 12.

Second, there is no indication that there are many of these back-end storage 1222, nor that any plurality of back-end storage 1222 includes a different one dedicated to each different one of the plural recoverable state management types. Thus, Apte does not teach providing the different back-end storages, e.g., an exemplary first back-end storage 1222 dedicated to the claimed disk replicated state management type and an exemplary second back-end storage 1222 dedicated to the claimed memory replicated state management type, as now claimed.

In support of this second assertion, reference is made to C17, L61-64, at which it is stated that:

EJB 1208 is not concerned with the manner in which the information is persisted-

EJB 1208 merely flags certain fields for persistence.

Clearly, this does not indicate that the Apte persistent state (recoverable) is to be classified. That is, the flags are generic, indicating only persistence and not persistence via a state server dedicated to state objects of a particular state. Thus, in terms of the claimed dedication, this C17 cite does not mean that Apte teaches replicating a state object corresponding to memory replicated state management type, in which the replicating is only to a back-end server 1222 that only stores the state objects that correspond to the memory replicated state management type. Similarly, this C17 cite does not mean that Apte teaches replicating a state object corresponding to disk replicated state management type, in which the replicating is only to a back-end server 1222 that only stores the state objects that correspond to the disk replicated state management type. Rather, only non-replicated (not persisted) and replicated (persisted) are indicated by the presence of a flag, and the flags indicate no claimed dedication of the back-end server 1222 in the claimed manner.

Stated in terms of the Apte details, with Apte teaching this generic persistence (via the field flags) and teaching the one back-end storage 1222, there is no basis in Apte on which to say that the back-end storage 1222 for a particular flagged field is dedicated to any particular type of persisted recovery. To the contrary, Apte does not define types of persistence, such that the persistence of all of the fields 1210, 1212, and 1214 is by storage in the one back-end storage 1222 (noted at C18, L3 as “1214”).

In addition to the C17-C18 descriptions of FIG. 12, for completeness as to “associated text”, it is noted that C2, L1-2 indicates that:

“Most back-end stores allow the persistence of primitive data types....”

Clearly, this is not a teaching that each of many “back-end stores...” is dedicated to one different state management type.

In view of these remarks, it is respectfully requested that this Subsection 2.2.C.1 cite be recognized as not teaching the claim clauses to which the cite was applied.

Subsection 2.2.C.2: Discussion of The Cite To:

“(see at least *server 104* FIG. 1 and associated text, *additional servers* col. 3: 58-col. 4:5”;

In re claim 9, the cite (Action page 10, top) to Apte was for a plurality of state servers “(see at least *server 104* FIG. 1 and associated text, *additional servers* col. 3: 58-col. 4:5)”. The cite relates to claim 9 text of:

replicating each particular state management unit in one of a plurality of state servers according to the particular state management type that corresponds to the particular state objects classified in the particular state management unit.

Referring to FIG. 1, and to the associated text at C3 and C4, the server 104 is a network server (L58), and (L66+) provides data, etc., to clients 108-112. There is no teaching in Apte relating to the server 104 functioning in the claimed manner of the plurality of state servers which are used in state management based separately on each different state management type, which type corresponds to the state objects associated with the different state management type (claim 1), for example. Further, there is no teaching in Apte relating to the server 104 functioning with an operation of replicating each particular state management unit in one of a plurality of servers 104 being a state server, and no teaching of replicating according to the particular state management type that corresponds to the particular state objects classified in the particular state management unit (claim 9). Again, there is no teaching in Apte relating to the server 104 functioning as a state server dedicated to each state management type, the state management type identifying a policy for replication of a state object to a state server dedicated to a particular state management type (claim 18).

It is respectfully requested that this *server 104* FIG. 1 cite be recognized as not teaching the claim clauses to which the cite was applied.

Subsection 2.2.C.3: Discussion of The Cite To:

“*additional servers* col. 3: 58-col. 4:5”;

At C3, L58, Apte refers to the server 104, which is distinguished in Section 2.2.C.2. See for example L58 and 66. The reference to “additional servers” is at C4, L3, and is a general description of what distributed data systems may include. As a result of this generality, there is no teaching in Apte of the claim clauses identified in the next-to-last paragraph of Section 2.2.C.2.

It is respectfully requested that this *additional server* col. 3: 58-col. 4:5 cite be recognized as not teaching the claim clauses to which the cite was applied.

Subsection 2.2.C.4: Discussion of The Cite To:

“(see at least *EJBs, protocol, particular server, mechanisms, persistent, container* col. 7: 25-55)”]],

As to this C7 cite, for accuracy the following text was taken from the USPTO image of Apte and is copied here. Apparently as indicators of how Apte assertedly teaches the claimed content, the rejection cited the following words in this passage at C7 of Apte: “**EJBs**”, “**protocol**”, “**particular server**”, “**mechanisms**”, “**persistence**”, and “**container**”. Those words are noted in bold below in the copy of Apte at C7, L25-55.

In the depicted example, two Java beans may be employed that implement the client object 400 and server object 402. What makes a bean different from a pure object is that it has an external interface, called the properties interface, which allows a tool to read what the component is supposed to do and hook it up to other beans and plug it into another environment. Two different types of beans may be used-JavaBeans and Enterprise JavaBeans (**EJB**). JavaBeans are intended to be local to a single process and are often visible at runtime. This visual component may be a button, list box, graphic or chart, for example, but it is not a requirement.

An **EJB** is a non-visual, remote object designed to run on a server and be invoked by clients. An **EJB** can be built from multiple, non-visual JavaBeans. **EJBs** are intended to live on one machine and be invoked remotely from another machine, and **EJBs** have a deployment descriptor that is intended as a description about the bean that can be read by a tool. **EJBs** are also platform independent and can be used on any platform that supports Java.

Server beans or **EJBs** are remotely executable components or business objects deployed on the server. **EJBs** have a **protocol** that allows them to be accessed remotely, and this **protocol** also allows them to be installed or deployed on a particular server. They have a set of **mechanisms** that allow them to delegate major

qualities of service, security, transactional behavior, concurrency (the ability to be accessed by more than one client at a time), and **persistence** (how their state can be saved) to the **container** in which they are placed on the **EJB** server. **EJBs** get their behavior from being installed in a **container**, which provide the different qualities of service. Any platform independent JavaBean can be adopted, through the use of a deployment tool, into a platform specific EJB that has the correct qualities of services available to meet the specific requirements of existing business systems and applications.

With this separation between client object 400 and server object 402, changes to various business logic within server object 402 may be performed without having to modify client bean 400. This is desirable because there may be thousands of clients that access a single server. In addition, these processes also may be applied to programs written in non-current programming languages, such as COBOL or to programs for which source code is unavailable. Dynamic changes to such programs may be made by creating an interface for the program to make the program compatible with an object-oriented programming system, such as Java.

It is respectfully submitted that the text associated with these above bolded words does not teach the claimed aspects of claim 1 (providing state management clause), or claim 9 (classifying and replicating clauses), or claim 18 (state server and replicated state manager) clauses. For example, Apte teaches that “**EJBs** have a **protocol** that allows them to be accessed remotely, and this **protocol** also allows them to be installed or deployed on a **particular server**.” Clearly, this is a teaching of EJBs, and not the claimed state objects or the claimed state management units. Further, the Apte reference to deployment of the EJBs on the cited “particular server” does not teach that the particular server conforms to the claimed “state management type identifying a policy for replication of a state object to a state server dedicated to a particular state management type”, for example.

As other examples, in Apte the EJBs are said to have a set of **mechanisms** that allow them to delegate major qualities to the **container** in which they are placed on the **EJB** server. But this reference to containers is general, and does not teach the details of the claim 1 and 18 limitation in re dedication of the state servers to a particular state management type. For example, Apte says: “**EJBs** get their behavior from being installed in a **container**, which provide the different qualities of service.” Clearly, provision of “different qualities of service” is not a teaching of a service quality that provides the claimed state objects, the claimed state management units, or the claimed state servers, each dedicated to a particular state management type, and is not a teaching of the claimed replication of a particular state management unit to the state server that is dedicated to the particular state management type of the particular state object.

With respect, the same comment in re generality applies to the Apte-taught words: “protocol”, and “persistence”.

In view of these remarks, it is respectfully submitted that Apte does not teach the claim 1, 9 or 18 limitations relating to state servers or replication. As to claims 1, and 4-5, because every claimed aspect of the defined operations must be shown by the reference cited for anticipation, and because Apte is shown to be lacking these noted features, withdrawal of the rejection of claims 1, and 4-5 based on anticipation by Apte is respectfully requested. As to independent claims 9 and 18, because the combined references must show every claimed aspect of the defined operations to make a *prima facie* case of obviousness, withdrawal of the rejections of claims 9, 13-17, 18-21, and 25 based on Section 103 is respectfully requested.

Subsection 2.2.D:

Subsection 2.2.D.1: Response In Re Claim 4:

By the above remarks it is seen that many identified limitations of independent claim 1, and thus also of dependent claims 4 and 5, are not taught by Apte. Therefore, it is respectfully submitted that the 35 USC Section 102 requirement as to anticipation has not been met by Apte, and withdrawal of the rejection is respectfully requested.

Additionally, as to claim 4, it is respectfully submitted that the description at C7, L25-55 does not meet the 35 USC Section 102 requirement as to anticipation. The rejection cited “(see at least *EJBs, container, protocol, persistence* col. 7: 25-55)”. Thus the rejection cited many individual words that are included in this cite. However, these words do not describe the claimed:

operation of grouping the state objects based on the type of state management to which the corresponding entity bean object is classified.

In detail, as noted above, the cite to C7 is to a passage that describes EJBs. It is respectfully submitted that the “separation” described here (C7, L55, separation between the Apte client object 400 and the Apte server object 402) is separation by the use of the container. This is not the claimed provision of state objects and then grouping those state objects based on state management type. Moreover, referencing C17 again, even if the fields 1210, 1212, and 1214 are considered as being state objects due to the EJB flagging certain fields (C17, L63), FIG. 12 makes it clear that Apte does not group the fields as claimed (i.e., based on the

type of state management to which the corresponding entity bean object is classified). In the FIG. 12 situation, the fields are either flagged or not flagged, and all flagged fields are stored to the one back-end storage 1222. Finally, as noted, Apte does not describe either of the claimed disk replicated state management type or memory replicated state management type, and thus there is no basis to say that Apte teaches grouping state objects based on the claimed state management types.

In further support, FIG. 12 shows no state objects and no state management units for the claimed grouping of state objects, and the description of FIG. 12 does not teach any such state objects or units. In comparison, Applicants' FIG. 3 shows the claimed grouping (see 308 and 310), and FIG. 3 shows replicated state management relating to groups 308 and 310 to provide the claimed state management based on state management type and the grouped state objects. Apte's failure to show such grouped state objects of many entity beans 1208, and failure to relate any such group to state management as in such FIG. 3, is further indication that Apte does not teach the claimed grouping of the state objects based on the type of state management to which the corresponding entity bean object is classified.

Therefore, it is respectfully submitted that the 35 USC Section 102 requirement as to anticipation of claim 4 has not been met by Apte, and withdrawal of the rejection of claim 4 is respectfully requested.

Subsection 2.2.D.2: Response In Re Claim 5: Additionally, as to claim 5, it is respectfully submitted that the Apte teaching at the cites to FIG. 12, and the description at C7, L25-55, do not meet the 35 USC Section 102 requirement as to anticipation. The rejection cited many individual words that are included in this cite. However, these words do not describe the claimed:

A method as recited in claim 4, wherein the state management type into which a group of state objects are grouped identifies a policy for replication of the group of state objects to the dedicated state server that is dedicated to the particular state management type corresponding to the group.

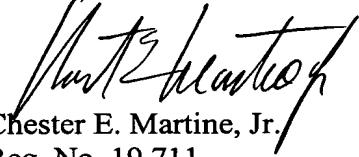
In detail, the above remarks in Section 2.2.D.1 in re claim 4 make it clear that Apte fails to teach the provision of the claimed the state management type into which a group of state objects are grouped. Further, the above remarks in Section 2.2C.indicate that Apte fails to teach the claim 5 text of "dedicated to the particular state management type corresponding to the group".

Application No. 09/825,249
Response dated April 17, 2006
Response to Action mailed January 17, 2006

Therefore, it is respectfully submitted that the 35 USC Section 102 requirement as to anticipation of claim 5 has not been met by Apte, and withdrawal of the rejection of claim 5 is respectfully requested

In view of the foregoing, Applicants respectfully request allowance of the pending claims, as may be amended herein. Accordingly, a notice of allowance is respectfully requested.

Respectfully submitted,
MARTINE PENILLA & GENCARELLA, LLP



Chester E. Martine, Jr.
Reg. No. 19,711

710 Lakeway Drive, Suite 200
Sunnyvale, California 94085
(408) 774-6927
Customer Number 32291